Tasks / Efficient Alarm Clock

Efficient Alarm Clock

In Pintos, threads may call this function to put themselves to sleep:

```
/**
* This function suspends execution of the calling thread until time has
* advanced by at least x timer ticks. Unless the system is otherwise idle, the
* thread need not wake up after exactly x ticks. Just put it on the ready queue
* after they have waited for the right number of ticks. The argument to
* timer_sleep() is expressed in timer ticks, not in milliseconds or any another
* unit. There are TIMER_FREQ timer ticks per second, where TIMER_FREQ is a
* constant defined in devices/timer.h (spoiler: it's 100 ticks per second).
*/
void timer_sleep (int64_t ticks);
```

`timer_sleep` is useful for threads that operate in real-time (e.g. for blinking the cursor once per second). The current implementation of `timer_sleep` is inefficient, because it calls `thread_yield` in a loop until enough time has passed. This consumes CPU cycles while the thread is waiting. Your task is to reimplement `timer_sleep` so that it executes efficiently without any busy waiting.